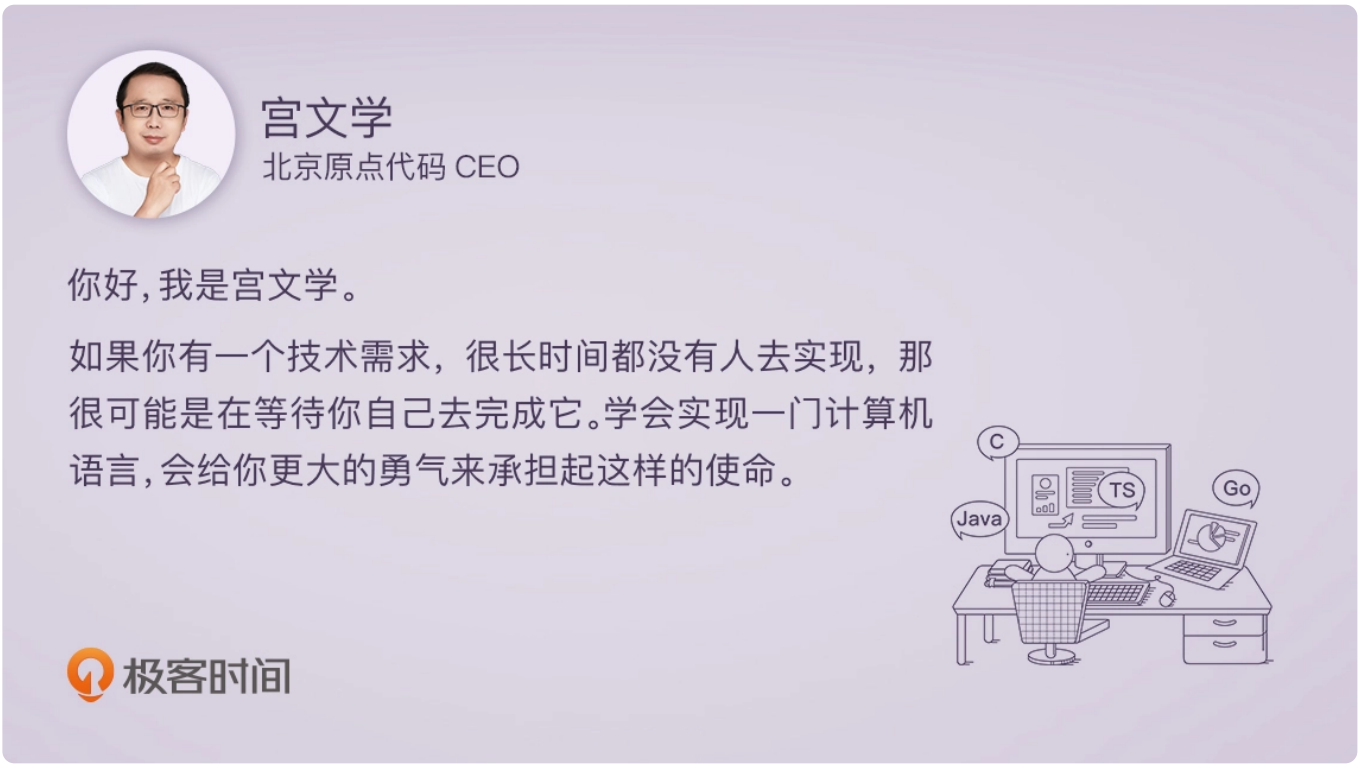
结束语｜等待你大展身手的那些领域

2021-11-24 宫文学

《手把手带你写一门编程语言》 课程介绍 >



讲述：宫文学
时长 14:21 大小 13.15M

你好，我是宫文学。

到今天为止，我们这门课的主要内容就都更新完了。不过，还有一些补充性的内容，我会通过加餐和开源项目的方式，继续和你保持沟通。

今天的结束语，我想跟你探讨一下，**学习实现一门语言的相关技术，到底会有什么用途。**

我会分成领域编程语言、平台级的软件和通用编程语言这三个话题，分析一下 编程语言技术能帮助你抓住哪些机会，让你有机会从普通的程序员进阶成大神级的程序员，并创造一些卓越的产品。

首先，我们来谈谈领域编程语言这个话题。

领域编程语言（DSL）

对于我们大部分同学来说，其实很难有机会，或者也没有这个意愿，去参与实现一门通用性编程语言。不过，其实在大部分情况下，我们也没有必要追求那么大的目标。有时候，针对我们所在的领域，实现一门领域编程语言，就是很有意义、很有成就感的事情。

我举几个我遇到的 DSL 的例子，看看能否抛转引玉，让你找到更多可以设计和使用 DSL 的场景。

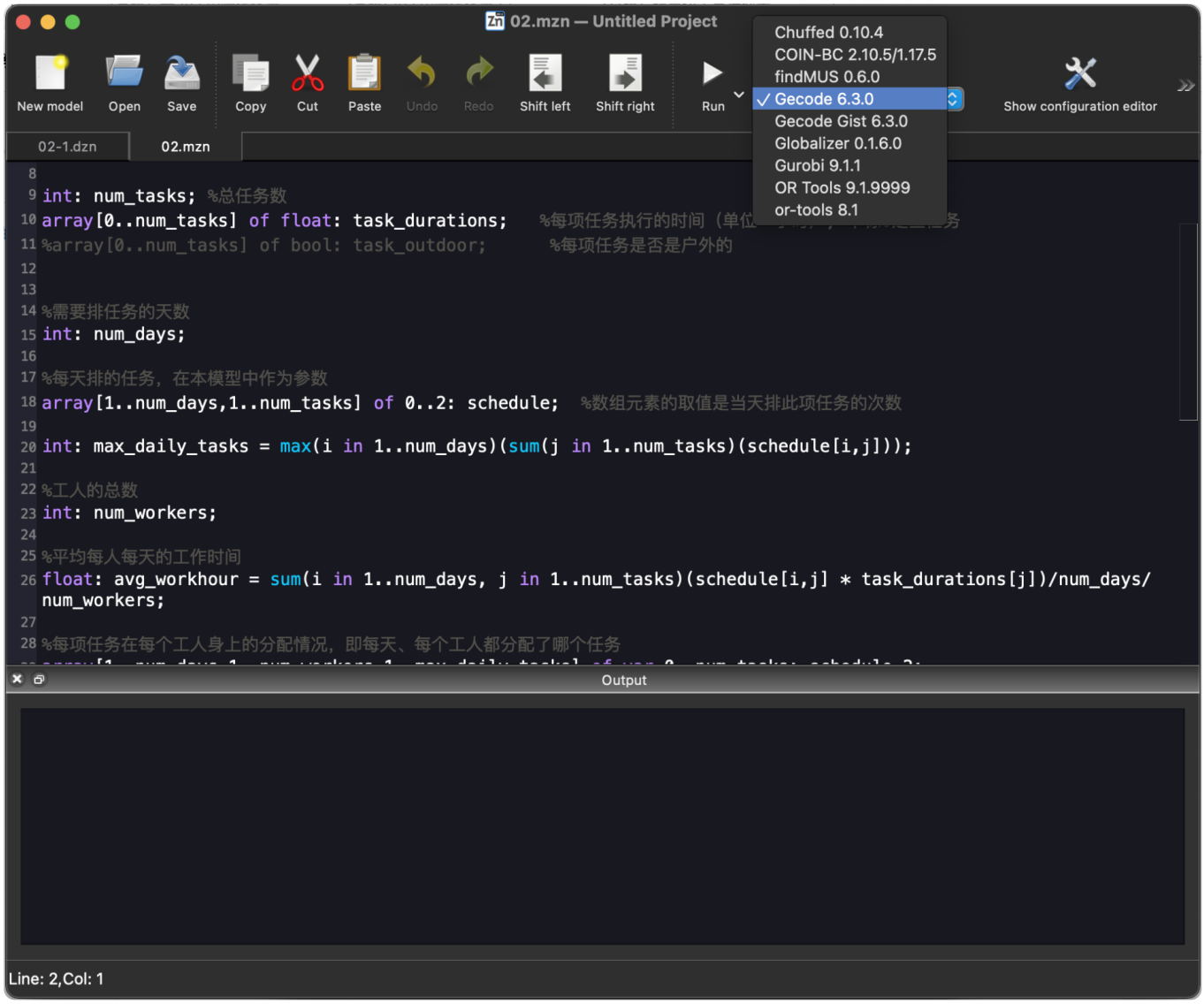
MiniZinc：最优化领域的开发工具

在 2020 年的 12 月，我曾经研究了一下最优化算法相关的技术和工具，看看它能否用于我们的一个产品。

很多同学在大学都学过最优化相关的理论，像线性规划、非线性规划这些，都属于这个领域。你也可能听说过运筹学，它们的意思差不多。最优化理论在实践中有很多用途。比如，我要解决一个应用问题，就是在某个领域，有很多员工，也有很多任务要完成。每个员工的技能是不同的，我需要通过算法来安排这些员工的工作，取得整体最优的效果。

为了实现最优化求解，有人开发了各种求解器，有商业的，也有开源的。但对于我一个新手来说，我一开始并不知道要用哪个工具，有点茫然。

通过某些途径，我了解到了 MiniZinc 这个工具。这个工具提供了一种 DSL，能够描述各种最优化问题，然后调用各种不同的求解器来求解。比如在下图，你能看到菜单栏有一个下拉菜单，里面有多个求解器。



MiniZinc 这个工具一下子解决了我的两个需求。首先，这个 DSL 很友好、很直观。你完全可以按照最优化的理论，描述一个问题的变量、参数、约束条件，然后就可以求解了，非常方便。第二，我暂时也不用关心不同的求解器的差别，可以随便选一个先用着，或者换着用不同的求解器，看看它们在性能和求解结果有哪些差异。

所以我很快就用 MiniZinc 编写了几个小程序来验证我的想法，并在较短的时间内取得了一些成果。

在使用这个工具的时候，我就在想，上过我这门课的同学，有没有能力做这么一个工具呢？我们来分析一下。

其实，要实现 MiniZinc，主要的工作就是**实现一个编译器的前端**，做词法分析、语法分析和语义分析工作。这个 DSL 的语法和语义都不是很复杂，所以工作量并不大。

做完前端的工作以后，程序就可以基于 AST 来解释执行了。解释执行的过程，其实就是调用各个求解器的 API，并把结果显示到界面上。

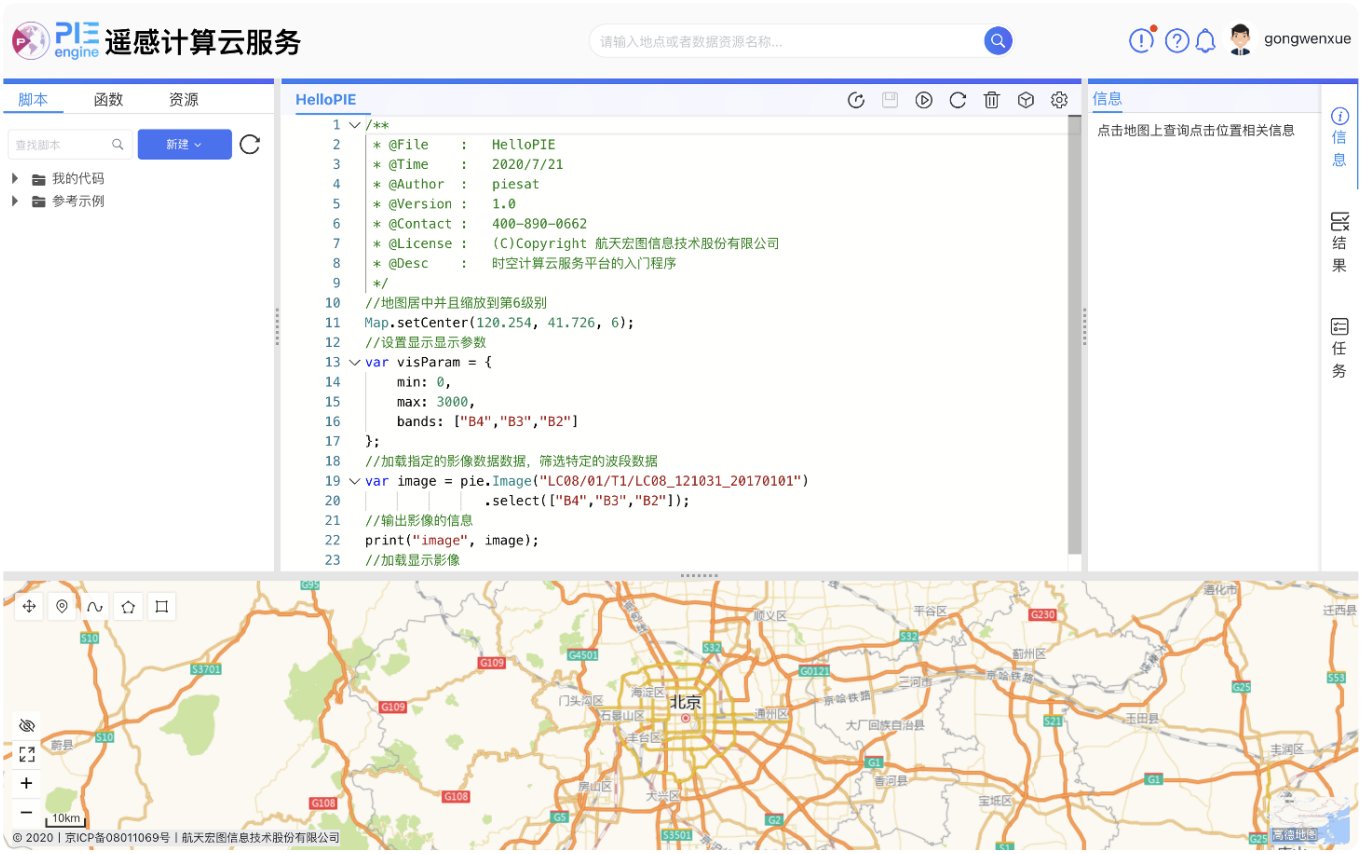
这么一个小工具，会给那些最优化领域的工作者和科研人员带来很大的便利。因为他们通常专注于研究算法，对于通用的计算机编程并不是很熟练。

那在你的领域中，是不是也有这样的情况呢？你是 IT 专业的人员，而你同事可能是其他专业的专家，比如是工程专家、投资专家、财务专家等等。你能否针对他们的领域，设计出一些 DSL，并提供一个开发工具来解决他们的一些痛点问题呢？

我们再看看第二个例子，这个例子是遥感领域的一个二次开发平台。

遥感领域的二次开发平台

我硕士的专业是在遥感和 GIS 领域，我有一个硕士同学在这个领域做出了一个上市公司。他们有一个产品，是一个遥感云平台，也就是把全国各地的很多遥感资料都放到云上管理。这个云平台里有一个开发工具，能够让用户在浏览器里编写程序，调用云平台的 API，实现遥感数据分析等功能。



据说这个平台是跟 Google Earth 对标的，国外很多科学家都会在 Google Earth 平台上写程序处理遥感数据，并构建自己领域的应用。

如果让你去实现这个开发工具，你会怎么做呢？

首先，你肯定需要**一个基于 Web 的代码编辑器**，这方面有好几个开源工具，所以这点并不是难题。

接下来，你仍然要实现**编译器前端的工作**。这一次，你需要编译的是 JavaScript 语言，它的语法特性和语义特性都比较多，所以实现的工作量要大一些。你可以把我们现在的词法分析器和语法分析器改一改，来实现 JavaScript 的解析。如果你想偷懒，还可以直接用 antlr 和现成的语法规则生成一下解析器。不过，无论如何，语义分析的工作是省不了的。你需要建立符号表、进行符号的消解，但不需要像我们这门课这样做那么多的类型处理。

完成编译器前端以后，还要做些什么呢？我们还要**做一些中端的优化工作**。因为这个开发工具要调用与遥感有关的 API。而如何调用这些 API 才会让效率最高呢？所以这一点上，我们实际上是要做一些优化的。

做完中端优化后，后续的编译和运行过程，有可能只要交给一个成熟的 JavaScript 引擎就可以了。

怎么样？使用我们这门课上学过的技术，你可以把很多科研工作者、各个行业的应用开发人员，都聚集到一个平台上，充分释放海量遥感数据的价值，这是不是一件挺酷的事情？

接下来，我再举一个 CAD 领域的例子，这个产品是 OpenCAD。

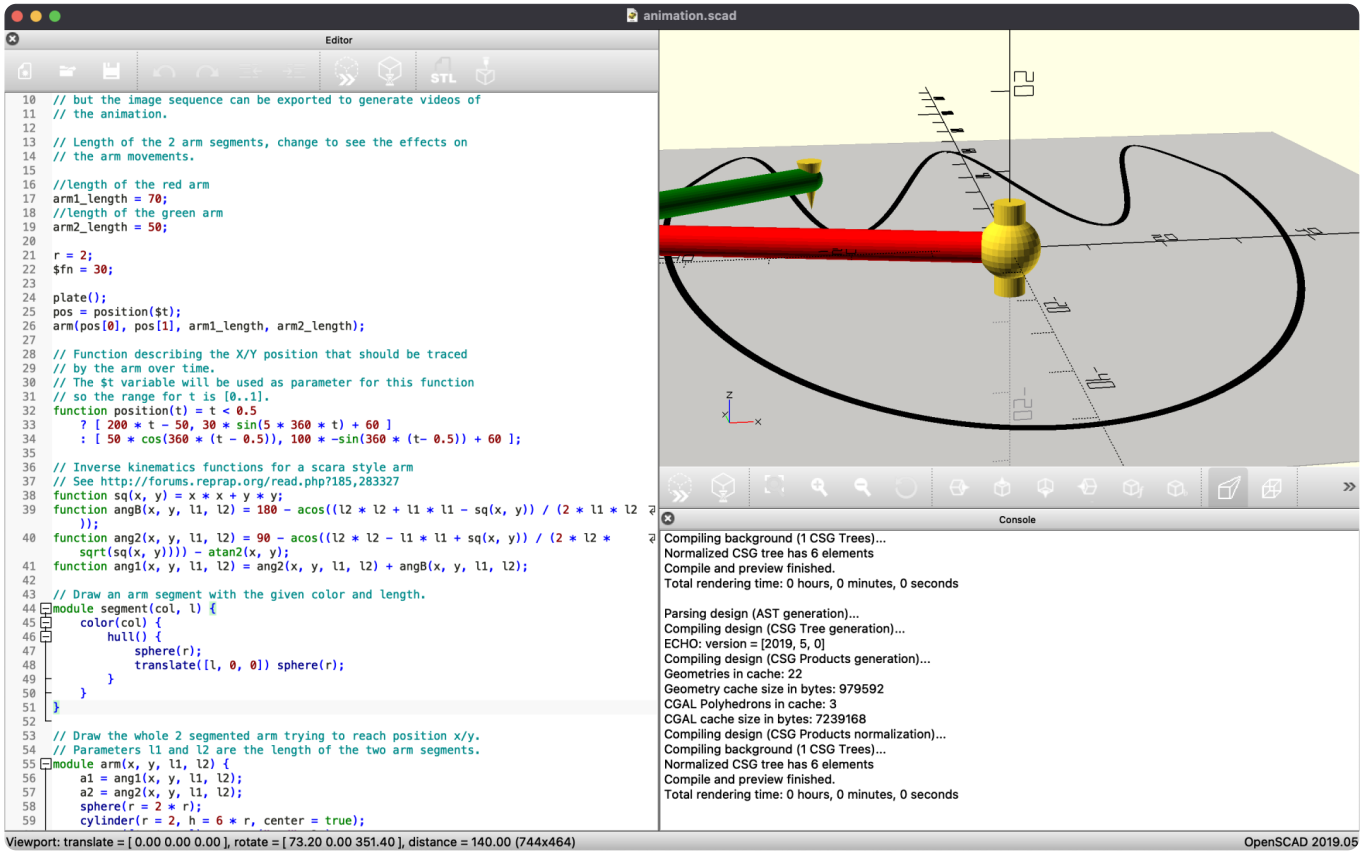
OpenCAD

几个月前，我一时兴起买了一台 3D 打印机，想自己打印点好玩的东西。但是在打印之前，还需要建立 3D 模型，所以我就搜了搜建模工具，发现了一个叫做 OpenCAD 的软件。

这个软件提供了一个编程界面。你可以在这个界面中，通过编程来创建长方体、圆柱体这样的三维对象，也可以通过编程来控制它们的位置、旋转的角度。

另外，你还可以创建模块，把多个基础的对象拼成复杂的对象，比如把一些长方体、圆柱体拼成一辆车。之后，这辆车就可以作为一个整体，用来构架更复杂的场景。

并且，模块还可以带参数，就像一个函数或者一个类那样。你通过调整模块的参数，就可以调整生成的 3D 对象。比如同样是一辆车，你可以通过调整参数生成一辆很大的车，也可以生成一辆玩具车。



我并不熟悉 CAD 领域，所以很难把这个软件跟其他 CAD 软件做客观地对比。不过，我能够看出，OpenCAD 通过编程来建模的形式，有几个特别的优势：

第一个优势是精准。在建立某些机械模型的时候，零件的大小、位置等信息必须是精准的。而使用编程语言，你可以用数学公式做出各种精准地计算。比如，上面的例子就使用了三角函数来精确的绘制曲线。

第二个优势是可重用性。可重用性是编程语言的基本特征，模块、函数、类，都是可重用的元素。并且，这种可重用的元素都是参数化的。在不同的场景中，可以通过调整参数来获得想要的功能。

所以，要开发一款优秀的 CAD 软件，我们需要充分吸收编程语言的技术。类似的领域还有建筑建模软件（BIM）、城市建模软件（CIM）等，现在很多公司的产品，都声称提供了低代码的数字城市平台，那这些产品也应该充分使用计算机语言的技术，并形成特定领域的 DSL。

你看，我们现在分析了三个案例，看到了三个领域对计算机语言的要求。其实，这样的领域还有很多，并且它们都可以受益于领域编程语言。我希望你能受到这些例子的启发，看看在自己的领域内还有没有这样的需求，说不定你会也能做出一些开创性的事情。因为这样级别的工作，必须在掌握了我们这门课的知识体系以后，你才有能力驾驭。

上面这些采用了 DSL 的软件，基本都属于一些平台级的软件。那我们再围绕如何实现平台级的软件这个话题，展开讨论一下编程语言技术的作用，和你当前面对的机会。

实现平台级的软件

Lisp 语言的重要推广者、《黑客与画家》的作者、创业孵化器 Y Combinator 的创始人保罗·格雷厄姆曾经说过一段话，大概就是说，每个软件演化到最后，都会内置一个 Lisp 语言的实现。

他的这段话，其实是说，任何软件，如果想覆盖尽量多的应用场景，都需要提供一定的编程能力。我举几个例子来说明一下这个观点：

为了更高效的管理很多服务器上的操作系统，搞运维的技术人员都会用 Python 来写脚本。这个时候，我们是在用脚本语言来扩展操作系统的功能；

数据库系统之所以能够满足各种应用的需求，是因为它是通过 SQL 语言来访问数据库的功能；

微软的各种应用产品，几乎都提供了二次开发的能力，这让你可以基于微软的产品形成各种不同领域的解决方案。比如，如果你能够熟练使用 Excel 里面的宏和编程技术，你可以完成很多的数据分析需求；

三维游戏引擎提供编程功能，让你能够创建各种三维游戏场景；

各种报表或 BI 工具，都提供了定义数据源、定义报表公式等功能，方便你设计各种报表；

工作流或 BPM 系统，需要提供流程设计、公式定义和自定义逻辑的功能，来满足各种不同的流程场景了；

对于一个 API 网关来说，需要提供一定的编程逻辑，来定义在什么情况下，把 API 访问路由到哪个微服务，或者进行熔断。

类似的例子还有很多。

如果你的软件只是为某个用户个性化定制的，只需要满足这一个客户的需求，那么你只要弄清楚需求，然后实现出来就行了。但如果你想让更多的用户使用你的软件，那该怎么办呢？

在少量情况下，你可以设计一套标准的软件，并让所有的用户都满意。比如，几乎所有的字处理软件的功能都差不多，你只需要购买一个 License 就行了，很少会提出个性化的需求。

但这样的通用软件是很少的。更多的情况下，特别是在企业应用领域，我们都需要对软件的标准功能做一定的调整，让它符合某个客户特定的需求。比如，你可能需要调整某些业务规则、某些流程、某些数据项，等等。

而国内大部分软件公司，目前都是通过修改源代码来满足这些个性化的需求。这就导致软件的实施成本很高，版本难以维护，这是很多应用软件开发商所处的困境。你在的公司，也可能存在这样的困境。

这个时候，如果你能把应用软件上升为平台，也就是在不修改原来的源代码的基础上，提供二次开发的能力，二次开发的部分由各个客户自行维护，才可以从根本上打破这种困境。从这个角度看，我们每个同学所在的领域，都存在着大量的潜在机会，采用这门课学习到的编程语言技术，你可以把你所在领域的软件，提升成一个平台级的软件。

还有最后一种情况，就是你本来就要开发平台级的软件，比如数据库系统、表单系统、报表系统、游戏引擎等，那么编程能力就更加是缺省的要求。

不管怎样，只要你想让你的软件变成平台级的软件，具备适应各种不同应用场景的能力，具备扩展功能的能力，你就需要采用这门课教给你的编程语言技术，对这些软件进行改

造。在我看来，国内有太多软件产品需要进行这种提升了，这都是你可以施展身手的机会。

好了，聊完了领域编程语言和平台级软件以后，你还有没有其他机会来大显身手呢？有的，这就是实现通用编程语言这个终极大 Boss。

通用编程语言

像 Java、C、Go 这些语言，能够用于很多领域，所以它们被叫做通用编程语言。到目前为止，我们国家还没有正式发布的、被广泛接受的通用编程语言。

不过信息灵通的同学可能也知道，我们多个大厂，其实都在内部酝酿和研发这样的语言。我本人也在参与某门语言的内测和评价工作。鉴于保密协议的约束，在这门语言没有正式发布前，我是不能谈论它的名称和技术细节的。

所以，如果你对实现通用编程语言很感兴趣的话，其实现在就有机会进入这些团队，贡献自己的一份力量。

如果你参与了这样的项目团队，那么你就可以选定一个具体的领域，深入研究我们这门课涉及的那些知识点。有人可能变成语法分析的专家，有人可能成为优化技术的专家，有人可能成为后端技术的专家，还有人可能会成为运行时方面的专家。

而我相信，**正是你们这些未来的专家，将来必然会让中国的通用编程语言领域大放异彩！**

写在最后

这门课程的主体内容，到这里就正式结束了，不过我还会跟同学们保持联系。保持联系的方式有几个，一是开源项目 PlayScript，一是计划不定期发布的几篇加餐，还有就是这门课的微信群。

我会继续在编程语言的领域探索和实践，希望能够跟你多多交流。也希望你在前进的道路上，能够找到更多的志同道合的朋友，一起砥砺前行，创造出优秀的作品！

另外，我还给你准备了一份毕业问卷，题目不多，希望你能在问卷里聊一聊你对这门课的看法。欢迎你点击下面的图片，用 1 ~ 2 分钟的时间填写一下。当然了，如果你对课程内

容还有什么问题，也欢迎你在留言区或交流群继续提问，我会持续回复你的留言。

我是宫文学，我们江湖再见。

分享给需要的人，Ta订阅后你可得 20 元现金奖励
生成海报并分享
赞 0 提建议
© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 期末考试丨实现编程语言这些核心知识点，你掌握得咋样了？

精选留言 (6)

写留言

He
2021-11-24

感谢老师分享~

展开

1

Geek_80559d
2021-11-24

哇哇，感谢老师的分享，收获了好多干货，拓展视野

展开

1

奥巴鸟
2021-11-24

3D游戏引擎很感兴趣，就是还不知道如何下手

展开

1

奋斗的蜗牛
2021-11-24

谢谢老师的分享，收获满满

展开

1

峰宇时空
2021-11-24

谢谢老师，开阔了眼界，扩展了选择。

展开

2

千无
2021-11-26

感谢宫文学老师，这一路，与君共勉

展开